

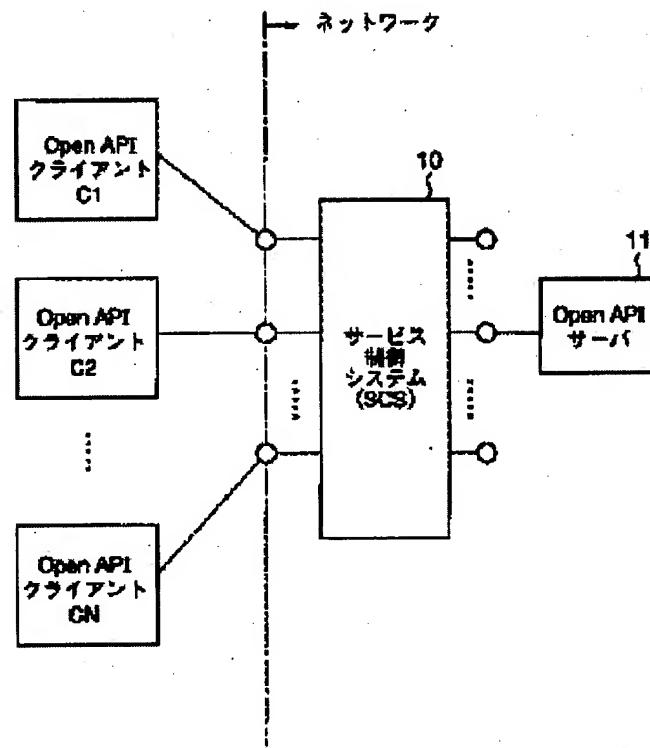
SERVICE CONTROL METHOD AND DEVICE IN OPEN API SYSTEM

Patent number: JP2003256223
Publication date: 2003-09-10
Inventor: SATO YASUSHI; EGUCHI YASUHISA; DANIEL A BERNET; FRAN PANKEN; BENNETT ANDREW JONATHAN
Applicant: NTT DOCOMO INC;; LUCENT TECHNOL INC
Classification:
 - **international:** G06F9/54
 - **European:**
Application number: JP20020061211 20020306
Priority number(s):

Abstract of JP2003256223

PROBLEM TO BE SOLVED: To provide a service controller and a service control method in an Open API system in which a plurality of service applications are simultaneously received by a telecom carrier network regardless of the service start conditions of those applications.

SOLUTION: A service control system 10 stores competing service start condition registration/change/deletion requests received from an Open API client in a memory, and analyzes the service start condition registration states from already stored service start condition registration/change/deletion requests. The received requests are compared with the service start condition registration states at that point of time so that the unchanged service start condition part, the service start condition part to be added, and the service start condition part to be deleted can be discriminated from each other, and the service start condition registration/change/deletion requests can be configured based on the discrimination, and noticed to an Open API server 11.



Data supplied from the esp@cenet database - Patent Abstracts of Japan

(19)日本国特許庁 (JP)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開2003-256223

(P2003-256223A)

(43)公開日 平成15年9月10日 (2003.9.10)

(51)Int.Cl.
G 0 6 F 9/54

識別記号

F I
G 0 6 F 9/06

テマコード(参考)
6 4 0 E 5 B 0 7 6

審査請求 未請求 請求項の数22 O L (全 10 頁)

(21)出願番号 特願2002-61211(P2002-61211)

(22)出願日 平成14年3月6日 (2002.3.6)

(71)出願人 392026693

株式会社エヌ・ティ・ティ・ドコモ
東京都千代田区永田町二丁目11番1号

(71)出願人 596092698

ルーセント テクノロジーズ インコーポ
レーテッド
アメリカ合衆国, 07974-0636 ニュージ
ャーシ, マレイ ヒル, マウンテン ア
ヴェニュー 600

(74)代理人 100097157

弁理士 桂木 雄二 (外1名)

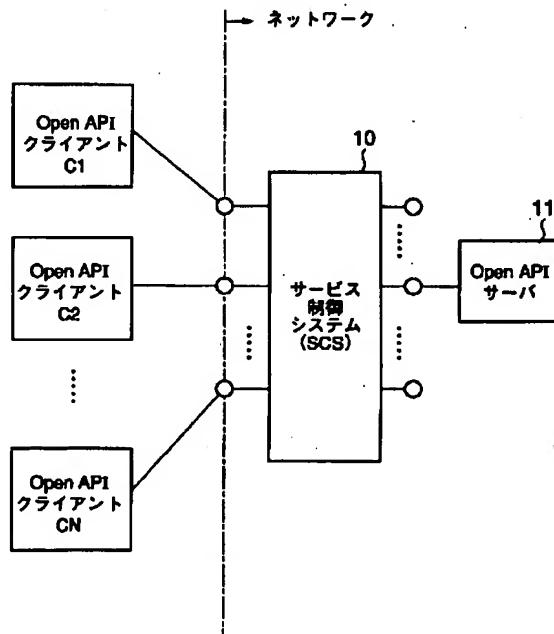
最終頁に続く

(54)【発明の名称】 オープンAPIシステムにおけるサービス制御方法および装置

(57)【要約】

【課題】 複数のサービスアプリケーションのサービス起動条件によらず、それらがテレコムキャリアネットワークに同時に受理されるOpenAPIシステムにおけるサービス制御装置及び方法を提供する。

【解決手段】 サービス制御システム10は、OpenAPIクライアントから受信した競合するサービス起動条件登録/変更/削除要求をメモリに格納し、既に格納されているサービス起動条件登録/変更/削除要求からサービス起動条件登録状態を解析する。受け入れた要求とその時点のサービス起動条件登録状態とを比較することで、変更が生じないサービス起動条件部分、追加すべきサービス起動条件部分、および、削除すべきサービス起動条件部分を区別し、その区別に基づいてサービス起動条件登録/変更/削除要求を構成しOpenAPIサーバ11へ通知する。



【特許請求の範囲】

【請求項1】 オープンAPI (Open Application Programming Interface) を利用したアプリケーション(以下、オープンAPIクライアントという。)と、テレコムネットワークの一部として動作するオープンAPIサーバと、を含むオープンAPIシステムにおいて、前記オープンAPIクライアントに対する前記オープンAPIサーバの代理として前記テレコムネットワーク内に設けられ、少なくとも1つのオープンAPIクライアントから受信したサービス起動条件登録/変更/削除要求を格納するメモリと、前記メモリに既に格納されている少なくとも1つのサービス起動条件登録/変更/削除要求からサービス起動条件登録状態を解析する解析手段と、新たに受信したサービス起動条件登録/変更/削除要求と前記サービス起動条件登録状態に基づいて、変更が生じないサービス起動条件部分と、変更すべきサービス起動条件部分と、削除すべきサービス起動条件部分とを区別する手段と、を有することを特徴とするサービス制御装置。

【請求項2】 前記オープンAPIサーバに対する前記オープンAPIクライアントの代理として、前記区別に基づいて前記オープンAPIサーバへ通知すべきか否かを判定し、通知すべき場合には、前記前記変更が生じないサービス起動条件部分と、前記変更すべきサービス起動条件部分と、前記削除すべきサービス起動条件部分とからサービス起動条件登録/変更/削除要求を構成し、前記オープンAPIサーバへ通知する通知判定手段を更に有することを特徴とする請求項1記載のサービス制御装置。

【請求項3】 オープンAPI (Open Application Programming Interface) を利用したアプリケーション(以下、オープンAPIクライアントという。)と、テレコムネットワークの一部として動作するオープンAPIサーバと、を含むオープンAPIシステムにおいて、前記オープンAPIサーバに対する前記オープンAPIクライアントの代理として前記テレコムネットワーク内に設けられ、

前記テレコムネットワーク上においてサービス起動条件が成立することで前記オープンAPIサーバからサービス起動条件成立通知を受信すると、当該サービス起動条件成立通知に対応するサービス起動条件登録を行った少なくとも1つのオープンAPIクライアントを特定する特定手段と、

前記特定手段により複数のオープンAPIクライアントが特定された場合には、それら全てのオープンAPIクライアントに対してサービス起動条件が成立した旨の通知を行う優先順位を決定する手段と、を有することを特徴とするサービス制御装置。

【請求項4】 前記オープンAPIクライアントに対する前記オープンAPIサーバの代理として、前記特定手段により1つのオープンAPIクライアントのみが特定された場合には前記サービス起動条件が成立した旨の通知を当該オープンAPIクライアントに送付し、前記特定手段により複数のオープンAPIクライアントが特定された場合には前記サービス起動条件が成立した旨の通知を前記優先順位に従って順次送付する通知手段を更に有することを特徴とする請求項3記載のサービス制御装置。

【請求項5】 前記通知手段は、1つのオープンAPIクライアントへ送付された前記サービス起動条件が成立了旨の通知に対して、当該オープンAPIクライアントから応答がない場合には、次の優先順位のオープンAPIクライアントへ前記サービス起動条件が成立了旨の通知を送付する、ことを特徴とする請求項4記載のサービス制御装置。

【請求項6】 前記オープンAPIクライアントおよびオープンAPIサーバは、何らインタフェースを変更することなく標準OpenAPI仕様のまま動作可能であることを特徴とする請求項1ないし7のいずれかに記載のサービス制御装置。

【請求項7】 オープンAPI (Open Application Programming Interface) を利用したアプリケーション(以下、オープンAPIクライアントという。)と、テレコムネットワークの一部として動作する複数のオープンAPIサーバと、を含むオープンAPIシステムにおいて、

前記オープンAPIクライアントに対する前記複数のオープンAPIサーバの代理として、前記オープンAPIクライアントからサービス起動条件登録/変更/削除要求および呼生成要求のいずれかを受信する受信手段と、前記複数のオープンAPIサーバに対する前記オープンAPIクライアントの代理として、前記受信手段により受信した要求を前記複数のオープンAPIサーバのいずれかに通知するAPI分配手段と、を有することを特徴とするサービス制御装置。

【請求項8】 前記複数のオープンAPIサーバは、前記テレコムネットワーク上に、

1) 物理的に複数だが論理的には一つ、
2) 物理的に複数で、論理的にも複数に分割される、
3) 前記1) および2) の組み合わせ、のいずれかのパターンで分散されており、前記API分配手段は、前記要求を分配すべき1つのオープンAPIサーバを決定することを特徴とする請求項7記載のサービス制御装置。

【請求項9】 オープンAPI (Open Application Programming Interface) を利用したアプリケーション(以下、オープンAPIクライアントという。)と、テレコムネットワークの一部として動作するオープンAPIサーバと、を含むオープンAPIシステムにおいて、

別するステップと、

を有することを特徴とするサービス制御プログラム。

【請求項17】前記オープンAPIサーバに対する前記オープンAPIクライアントの代理として、前記区別に基づいて前記オープンAPIサーバへ通知すべきか否かを判定するステップと、通知すべき場合には、前記前記変更が生じないサービス起動条件部分と、前記変更すべきサービス起動条件部分と、前記削除すべきサービス起動条件部分とからサービス起動条件登録/変更/削除要求を構成するステップと、前記構成されたサービス起動条件登録/変更/削除要求を前記オープンAPIサーバへ通知するステップと、を更に有することを特徴とする請求項16記載のサービス制御プログラム。

【請求項18】オープンAPI (Open Application Programming Interface) を利用したアプリケーション (以下、オープンAPIクライアントという。)と、テレコムネットワークの一部として動作するオープンAPIサーバと、を含むオープンAPIシステムにおける、コンピュータを、前記オープンAPIサーバに対する前記オープンAPIクライアントの代理として機能させるサービス制御プログラムにおいて、前記テレコムネットワーク上においてサービス起動条件が成立することで前記オープンAPIサーバからサービス起動条件成立通知を受信するステップと、当該サービス起動条件成立通知に対応するサービス起動条件登録を行った少なくとも1つのオープンAPIクライアントを特定するステップと、

前記複数のオープンAPIクライアントが特定された場合には、それら全てのオープンAPIクライアントに対してサービス起動条件が成立した旨の通知を行う優先順位を決定するステップと、

を有することを特徴とするサービス制御プログラム。

【請求項19】1つのオープンAPIクライアントのみが特定された場合には前記サービス起動条件が成立した旨の通知を当該オープンAPIクライアントに送付するステップと、

前記複数のオープンAPIクライアントが特定された場合には前記サービス起動条件が成立した旨の通知を前記優先順位に従って順次送付するステップと、

を更に有し、前記オープンAPIクライアントに対する前記オープンAPIサーバの代理として機能させることを特徴とする請求項18記載のサービス制御プログラム。

【請求項20】1つのオープンAPIクライアントへ送付された前記サービス起動条件が成立した旨の通知に対して、当該オープンAPIクライアントから応答がない場合には、次の優先順位のオープンAPIクライアントへ前記サービス起動条件が成立した旨の通知を送付するステップを含むことを特徴とする請求項19記載のサ

ービス制御プログラム。

【請求項21】オープンAPI (Open Application Programming Interface) を利用したアプリケーション (以下、オープンAPIクライアントという。)と、テレコムネットワークの一部として動作する複数のオープンAPIサーバと、を含むオープンAPIシステムにおける、コンピュータを、前記オープンAPIクライアントに対する前記複数のオープンAPIサーバの代理として機能させるサービス制御プログラムにおいて、前記オープンAPIクライアントからサービス起動条件登録/変更/削除要求および呼生成要求のいずれかを受信するステップと、前記複数のオープンAPIサーバに対する前記オープンAPIクライアントの代理として、前記受信した要求を前記複数のオープンAPIサーバのいずれかに通知するステップと、

を有することを特徴とするサービス制御プログラム。

【請求項22】前記複数のオープンAPIサーバは、前記テレコムネットワーク上に、

1) 物理的に複数だが論理的には一つ、
2) 物理的に複数で、論理的にも複数に分割される、
3) 前記1) および2) の組み合わせ、
のいずれかのパターンで分散されており、前記要求を分配すべき1つのオープンAPIサーバが決定することを特徴とする請求項21記載のサービス制御プログラム。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、テレコムキャリアネットワークが持つ呼制御機能やメール機能などのネットワーク機能を外部に公開することで、テレコムキャリア以外のものがアクセスして付加的なサービスアプリケーションの開発を可能にすることを目的としたオープンAPI (Open Application Programming Interface、以下、OpenAPIと記す。) システムに関する。

【0002】特に、本発明は、OpenAPIを利用したアプリケーションに関して複数のサービスアプリケーションが同時に動作できるように調停を行うサービス制御装置および方法に関する。

【0003】更に、本発明は、OpenAPIを利用したアプリケーションに対して、複数存在するOpenAPIサーバの分散構成を隠蔽する仕組みを提供するサービス制御装置および方法に関する。

【0004】

【従来の技術】テレコムサービスのユーザーがサービスを利用する場合、サービス提供条件が同じか、もしくは似通ったサービスを利用するケースを想定することができる。例えば、留守番電話サービスと転送電話サービスのようなものである。その際、テレコムサービスの提供者であるテレコムキャリアは、該当ユーザーが利用する

複数のサービスの機能や提供条件を解析し、最も適切なサービスを優先的に提供する必要がある。従来、その判定は全てのサービスの機能や提供条件を事前に把握した上でネットワーク上の交換機に固定的に作り込んでおく方法が取られていた。

【0005】このような固定的に作り込む方法に対し、OpenAPIでは、サービスを利用したアプリケーションがテレコムキャリアネットワークに対して自アプリケーションを起動せしめる条件を動的に事前に通知可能な仕組みが提供されている。この仕組みを利用して、OpenAPIでは、事前に通知された起動条件が満たされた時に、テレコムネットワークがそのサービスを起動する。したがって、テレコムネットワーク上には最も適切なサービスを優先的に提供するための固定的な判定処理が不要となる一方で、複数のサービスアプリケーションから動的に通知されたサービス起動条件を逐次解析し、最も適切なサービスを優先的に提供するための判定処理が必要となる。

【0006】また、OpenAPI仕様では、OpenAPIアプリケーションとOpenAPIサーバとがお互いを1対1に認識する必要がある。そのため、テレコムキャリアネットワークに複数のOpenAPIサーバが存在する場合には、OpenAPIアプリケーションは複数のOpenAPIサーバを全て個々に認識しなければならない。また、OpenAPIアプリケーションがOpenAPIサーバにAPIを用いて要求を通知する際に、その要求を複数のOpenAPIサーバのうちのどのサーバに通知するかを適切に決定しなければならない。

【0007】

【発明が解決しようとする課題】上述したように、OpenAPIでは、同じサービス起動条件を有するサービスアプリケーションはテレコムキャリアネットワークに一つしか登録することができない、という制限がある。例えば、留守番電話サービスと転送電話サービスについていえば、いずれか一つしかテレコムキャリアネットワークにサービス起動条件を登録することができない。このような制限下では、多くのサービスを柔軟に運用することができない。

【0008】また、テレコムキャリアネットワークに複数のOpenAPIサーバが存在する場合には、OpenAPIアプリケーションが適切なOpenAPIサーバを決定するためにOpenAPIサーバの分散構成を知る必要がある。そのことはOpenAPIアプリケーションの持つサービスロジックの複雑化を招き、サービス開発の効率化を阻害する。

【0009】そこで、本発明の目的は、複数のサービスアプリケーションがどのようなサービス起動条件を持っているとも、テレコムキャリアネットワークに同時に受理されるOpenAPIシステムにおけるサービル制

御装置及び方法を提供することにある。

【0010】本発明の他の目的は、複数のサービスアプリケーションに適切なサービスを優先的に提供可能なOpenAPIシステムにおけるサービル制御装置及び方法を提供することにある。

【0011】本発明の更に他の目的は、OpenAPIを利用したアプリケーションが、複数のOpenAPIサーバから構成されるテレコムキャリアネットワーク上で動作する場合において、OpenAPIサーバの分散構成を知ることなくサービス提供を可能とするOpenAPIシステムにおけるサービル制御装置及び方法を提供することにある。

【0012】

【課題を解決するための手段】上記目的を達成するために、オープンAPIネットワーク内に、オープンAPIクライアントおよびサーバの両者が使用可能でアクセス可能なネットワーク要素が必要である。このネットワーク要素はオープンAPIクライアントおよびサーバの両者と通信可能であることが必要であるが、これらが当該ネットワーク要素と通信していることを知るべきではない。OpenAPI仕様を強化するために、あるいはその不完全性のいくつかを隠すために設けられる全ての追加機能は、このネットワーク要素内にインプリメントされる。

【0013】本発明の第1の観点によるサービス制御装置は、オープンAPI (Open Application Programming Interface) を利用したアプリケーション(以下、オープンAPIクライアントという。)と、テレコムネットワークの一部として動作するオープンAPIサーバと、を含むオープンAPIシステムにおいて、前記オープンAPIクライアントに対する前記オープンAPIサーバの代理として前記テレコムネットワーク内に設けられ、前記オープンAPIクライアントから受信したサービス起動条件登録/変更/削除要求を格納するメモリと、前記メモリに既に格納されている少なくとも1つのサービス起動条件登録/変更/削除要求からサービス起動条件登録状態を解析する解析手段と、受信したサービス起動条件登録/変更/削除要求と前記サービス起動条件登録状態に基づいて、変更が生じないサービス起動条件部分と、変更すべきサービス起動条件部分と、削除すべきサービス起動条件部分とを区別する手段と、を有することを特徴とする。

【0014】前記オープンAPIサーバに対する前記オープンAPIクライアントの代理として、前記区分に基づいて前記オープンAPIサーバへ通知すべきか否かを判定し、通知すべき場合には、前記前記変更が生じないサービス起動条件部分と、前記変更すべきサービス起動条件部分と、前記削除すべきサービス起動条件部分とからサービス起動条件登録/変更/削除要求を構成し、前記オープンAPIサーバへ通知する通知判定手段を更に有

することを特徴とする。

【0015】本発明によるサービス制御装置は、前記オープンAPIサーバに対する前記オープンAPIクライアントの代理として前記テレコムネットワーク内に設けられ、前記テレコムネットワーク上においてサービス起動条件が成立することで前記オープンAPIサーバからサービス起動条件成立通知を受信すると、当該サービス起動条件成立通知に対応するサービス起動条件登録を行った少なくとも1つのオープンAPIクライアントを特定する特定手段と、前記特定手段により複数のオープンAPIクライアントが特定された場合には、それら全てのオープンAPIクライアントに対してサービス起動条件が成立した旨の通知を行う優先順位を決定する手段と、を有することを特徴とする。

【0016】前記オープンAPIクライアントに対する前記オープンAPIサーバの代理として、前記特定手段により1つのオープンAPIクライアントのみが特定された場合には前記サービス起動条件が成立した旨の通知を当該オープンAPIクライアントに送付し、前記特定手段により複数のオープンAPIクライアントが特定された場合には前記サービス起動条件が成立した旨の通知を前記優先順位に従って順次送付する通知手段を更に有する。

【0017】前記通知手段は、1つのオープンAPIクライアントへ送付された前記サービス起動条件が成立した旨の通知に対して、当該オープンAPIクライアントから応答がない場合には、次の優先順位のオープンAPIクライアントへ前記サービス起動条件が成立した旨の通知を送付することを特徴とする。

【0018】前記オープンAPIクライアントおよびオープンAPIサーバは、何らインタフェースを変更することなく標準OpenAPI仕様のまま代理装置として動作することを特徴とする。

【0019】上述したように、標準OpenAPI仕様に準拠したサーバおよびクライアントアプリケーションから構成されるシステムに、本発明によるサービス制御装置を加えることで、標準OpenAPI仕様では実現できない、多くのサービスアプリケーションが、どのようなサービス起動条件を持っていようと、テレコムキャリアネットワークに、全てのサービスアプリケーションが同時に登録可能となる。また、最も適切なサービスを優先的に提供するための判定を行うことが可能となる。

【0020】本発明の第2の観点によるサービス制御装置は、オープンAPI(Open Application Programming Interface)を利用したアプリケーション(以下、オープンAPIクライアントという。)と、テレコムネットワークの一部として動作する複数のオープンAPIサーバと、を含むオープンAPIシステムにおいて、前記オープンAPIクライアントに対する前記複数のオープン

APIサーバの代理として、前記オープンAPIクライアントからサービス起動条件登録/変更/削除要求および呼生成要求のいずれかを受信する受信手段と、前記複数のオープンAPIサーバに対する前記オープンAPIクライアントの代理として、前記受信手段により受信した要求を前記複数のオープンAPIサーバのいずれかに通知するAPI分配手段と、を有することを特徴とする。

【0021】前記複数のオープンAPIサーバは、前記テレコムネットワーク上に、1)物理的に複数だが論理的には一つ、2)物理的に複数で、論理的に複数に分割される、3)前記1)および2)の組み合わせ、のいずれかのパターンで分散されており、前記API分配手段は、前記要求を分配すべき1つのオープンAPIサーバを決定することを特徴とする。

【0022】上述したように、標準OpenAPI仕様に準拠したサーバおよびクライアントアプリケーションから構成されるシステムに、本発明によるサービス制御システムを加えることで、OpenAPIクライアントとOpenAPIサーバの間の静的な関係を切り離すことが可能となる。このことは、とりわけ、あるOpenAPIクライアントからの要求が1つのOpenAPIサーバへ転送されても、同じOpenAPIクライアントからの次の要求は別のOpenAPIサーバへ転送されるかもしれない、ということを意味する。サービス制御システムは、全てのOpenAPIクライアントに対してネットワークトポロジーを隠蔽する。

【0023】

【発明の実施の形態】以下、本発明の実施形態を図面を参照しながら詳細に説明する。

【0024】第1実施形態

1. 1) サービス制御システム

図1は、本発明の一実施形態によるサービス制御システム(SCS)を用いたOpenAPIシステムの概略的構成図である。ここでは、ネットワーク側にサービス制御システム10とOpenAPIサーバ11とが設けられ、サービス制御システム10が複数のOpenAPIクライアント(OpenAPIを利用したアプリケーション)C1~CNとOpenAPIサーバ11との間に介在している。サービス制御システム10のインタフェースは、OpenAPI仕様に準拠したインタフェースである。OpenAPIクライアントC1~CNもOpenAPIサーバ11もサービス制御システム10の存在を知る必要はない。

【0025】上述したように、OpenAPI(たとえば、Parlay)では、クライアントとサーバが1対1対応で動作する際のインタフェースのみが規定され、1つのクライアントは、1つの呼に対する制御を要求することができる。このようなシステムでは、複数のクライアントが相手の存在を知らないことがあるから、1つの呼の制御に対して複数の異なるクライアントが互いに

競合する要求を行う可能性がある。OpenAPIシステムではこのような競合する要求は禁止されるが、本発明によるサービル制御システム10によって、複数のクライアントが同じOpenAPI仕様を用いて同一呼の制御を、エラーを受信することなく、要求することが可能となる。

【0026】サービス制御プログラムは、ミドルウェアとして、一連のインターロック・ソフトウェア・モジュールから構成され、プラットホームに依存しないコードで書くことができる。たとえば、後述するように、サービス制御プログラムは、サービス起動条件登録/変更/削除要求を受け付けて格納するモジュール、既に格納されている要求を解析するモジュール、新たに受け付けた要求と現時点での解析結果とを比較してサーバへ通知する要求を作成するモジュール、および、サーバから要求成立通知を受けたときに、どのクライアントへどのような順番で要求成立通知を送信するかを決定するモジュールなどを含んでいる。

【0027】なお、サービス制御プロセスは、OpenAPIクライアントC1～CNとOpenAPIサーバ11との間で、通常のトランスポートシステム、すなわち、OpenAPIクライアントおよびサーバが互いの通信に現在使用しているトランスポートシステム（たとえば、CORBA、WSDLなど）を通して通信する。サービス制御ソフトウェア上で使用されるインターフェースは、すべてOpenAPI仕様に準拠したものである。

【0028】1. 2) 複数クライアントからの要求の受入および処理

図2は、本発明の一実施形態によるサービス制御システムの動作を説明するためのシーケンス図である。サービス制御システム10は、OpenAPIクライアントからサービス起動条件登録/変更/削除要求を受け入れ、処理すると、それらを適切にOpenAPIサーバ11へ送信する。要求が最初にクライアントから受信されたならば、処理された要求はOpenAPIサーバ11へ送信される。

【0029】1. 3) 複数クライアントへの要求成立通知

1つの呼の制御に対して複数のクライアントからの要求を受け付けた場合、サービス制御システム10は、クライアントの要求を満たす呼がネットワークから得られた時に、どのクライアントがその通知を受けるべきかを決定する必要がある。

【0030】図2において、ネットワーク上でサービス起動条件が成立した場合、サービス制御システム10はネットワークイベント（ここでは、要求成立通知11.0）を受信する。要求成立通知11.0を受信すると、サービス制御システム10はどのクライアントが最初に当該要求成立通知11.0を受け取るべきかを決定する

（ステップS12）。

【0031】OpenAPI仕様によれば、イベントを受信したクライアントは何らかの応答をするようになっている。したがって、要求成立通知11.0を受信すると、OpenAPIクライアントは応答をかえてくるはずである。もし何ら応答がない場合には（たとえば、クライアントのロジック障害、ネットワークの障害などが原因で）、サービス制御システム10は同じイベント（要求成立通知11.0）を次の優先順位のクライアントへ送付する。

【0032】もあるイベントに対してどのクライアントも受信することができないときには、サービス制御システム10はOpenAPIサーバ11へエラーを通知する。

【0033】第2実施形態

2. 1) サービス制御システム

図3は、本発明の他の実施形態によるサービス制御システム（SCS）を用いたOpenAPIシステムの概略的構成図である。ここでは、ネットワーク側にサービス制御システム20と複数のOpenAPIサーバS1～SMとが設けられている。サービス制御システム20は、OpenAPIクライアント（OpenAPIを利用したアプリケーション）21と複数のOpenAPIサーバS1～SMとの間に介在している。

【0034】サービス制御システム20はOpenAPIクライアント21とOpenAPIサーバとの間のプロキシ（代理装置）としてはたらく。サービス制御システム20は、OpenAPIクライアント21からイベント条件の登録/変更/削除要求あるいは呼生成要求を受け付けることができる。これらの要求は、後述するように分配先として適当なOpenAPIサーバが決定され、送信される。

【0035】OpenAPIクライアントからは、サービス制御システム20はOpenAPIサーバとして見える。OpenAPIサーバからは、サービス制御システム20は、

1) OpenAPIクライアントが認証された後、フレームワークに代わってサービスインターフェースを要求するフレームワークエンティティとして見える、あるいは、

2) たとえば他のOpenAPIクライアントに代わって呼生成を要求するOpenAPIクライアントとしてみえる。

【0036】サービス制御システム20のインターフェースは、OpenAPI仕様に準拠したインターフェースである。OpenAPIクライアント21もOpenAPIサーバS1～SMもサービス制御システム20の存在を知る必要はない。

【0037】図4は、サービス制御システム20の概略的ハードウェア構成図である。ネットワーク接続のため

のインターフェース201および204のほかに、本発明によるAPI分配制御プログラムを実行するためのプロセッサ202とサービス起動条件などの必要な基準データを格納するメモリ203とを少なくとも有する。

【0038】API分配制御プログラムは、ミドルウェアとして、一連のインターロック・ソフトウェア・モジュールから構成され、プラットホームに依存しないコードで書くことができる。たとえば、後述するように、API分配制御プログラムは、サービス起動条件登録/変更/削除要求あるいは呼生成要求を受け付けて格納するモジュール、サービス起動条件を参照することでどのOpenAPIサーバへ分配するかを決定するモジュールなどを含んでいる。

【0039】なお、API分配制御プロセスは、OpenAPIクライアント21とOpenAPIサーバS1～SMとの間で、通常のトランSPORTシステム、すなわち、OpenAPIクライアントおよびサーバが互いの通信に現在使用しているトランSPORTシステム（たとえば、CORBA、WSDLなど）を通して通信する。API分配制御ソフトウェア上で使用されるインターフェースは、すべてOpenAPI仕様に準拠したものである。

【0040】2. 2) API分配制御

OpenAPIクライアント21から受信した、呼および/または呼生成の要求を制御するための登録/変更/削除要求は、従来では、特定のOpenAPIサーバへ送付され得るだけである。このために、OpenAPIクライアントとOpenAPIサーバとの間の静的なマッピングは、実行中に変更することができない。たとえば、あるイベント条件が発生したかどうかを知らせてほしいというOpenAPIクライアントの要求に基づくと、この静的な関係は望ましいものではなくなる。このような不都合は、本実施形態によるサービス制御システムを導入することによって払拭され得る。

【0041】図5は、本発明によるサービス制御システムのAPI分配制御動作の一例を示すシーケンス図である。サービス制御システム20は、OpenAPIクライアント21からイベント条件の登録/変更/削除要求RQ21. 1あるいは呼生成要求RQ21. 2を受け付けることができる。

【0042】上述したように、OpenAPIクライアント21からは、サービス制御システム20はOpenAPIサーバとして見えるから、OpenAPIサーバのネットワーク構成はクライアントには隠されている。サービス制御システム20は、OpenAPIサーバの

ネットワークにクライアントとして登録されている。

【0043】

【発明の効果】以上説明したように、標準OpenAPI仕様に準拠したサーバおよびクライアントアプリケーションから構成されるシステムに、本発明の第1実施形態によるサービス制御システムを加えることで、標準OpenAPI仕様では実現できない、多くのサービスアプリケーションが、どのようなサービス起動条件を持っているとも、テレコムキャリアネットワークに、全てのサービスアプリケーションが同時に登録可能となる。また、最も適切なサービスを優先的に提供するための判定を行うことが可能となる。

【0044】さらに、標準OpenAPI仕様に準拠したサーバおよびクライアントアプリケーションから構成されるシステムに、本発明の第2実施形態によるサービス制御システムを加えることで、OpenAPIクライアントとOpenAPIサーバの間の静的な関係を切り離すことが可能となる。このことは、とりわけ、あるOpenAPIクライアントからの要求が1つのOpenAPIサーバへ転送されても、同じOpenAPIクライアントからの次の要求は別のOpenAPIサーバへ転送されるかもしれない、ということを意味する。サービス制御システムは、全てのOpenAPIクライアントに対してネットワークトポロジーを隠蔽する。

【図面の簡単な説明】

【図1】本発明の一実施形態によるサービス制御システム（SCS）を用いたOpenAPIシステムの概略的構成図である。

【図2】本発明の一実施形態によるサービス制御システムの動作を説明するためのシーケンス図である。

【図3】本発明の他の実施形態によるサービス制御システム（SCS）を用いたOpenAPIシステムの概略的構成図である。

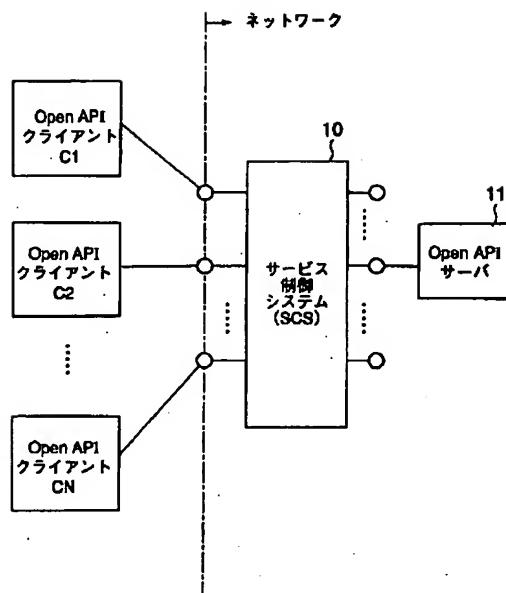
【図4】サービス制御システム20の概略的ハードウェア構成図である。

【図5】本発明によるサービス制御システムのAPI分配制御動作の一例を示すシーケンス図である。

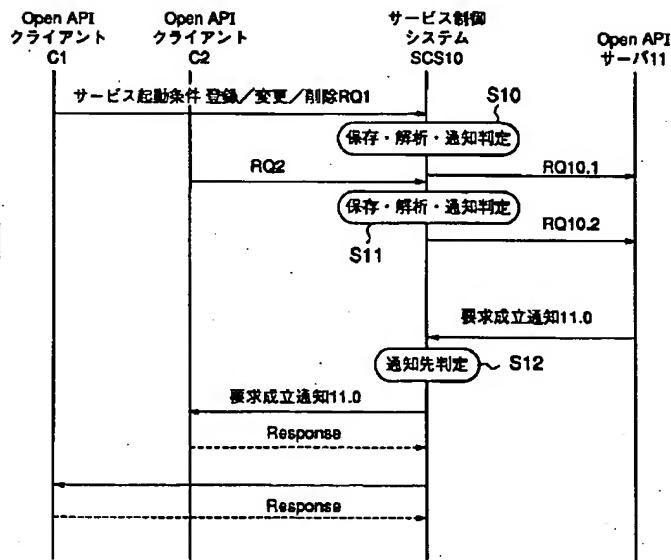
【符号の説明】

- 10 サービス制御システム
- 11 OpenAPIサーバ
- C1～CN OpenAPIクライアント
- 20 サービス制御システム
- 21 OpenAPIクライアント
- S1～SM OpenAPIサーバ

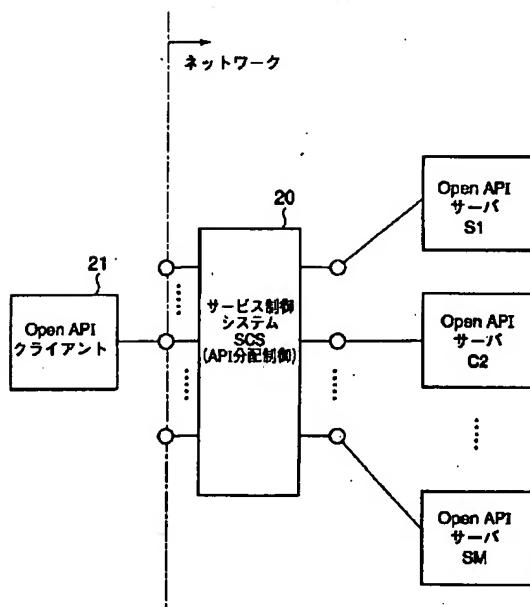
【図1】



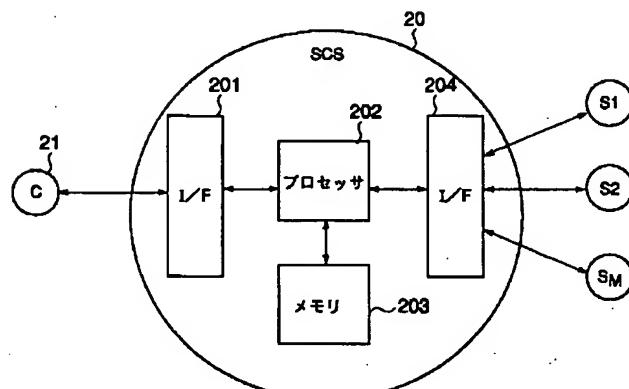
【図2】



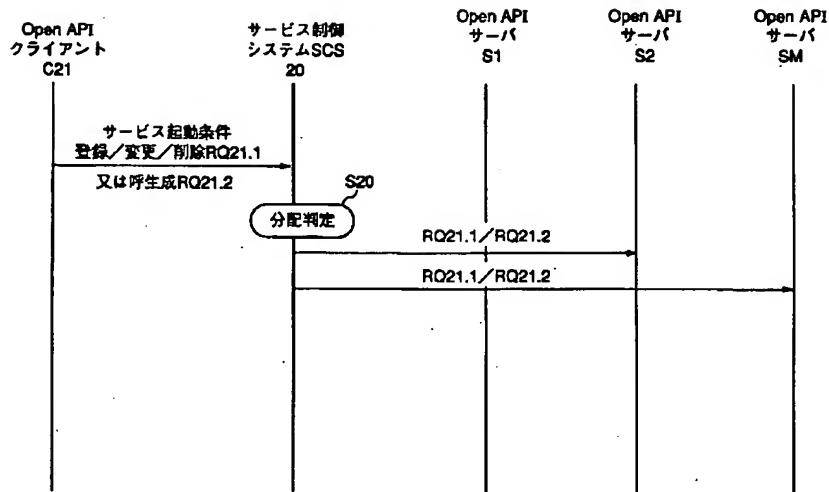
【図3】



【図4】



【図5】



フロントページの続き

(72)発明者 佐藤 恭
 東京都千代田区永田町二丁目11番1号 株式会社エヌ・ティ・ティ・ドコモ内
 (72)発明者 江口 泰央
 東京都千代田区永田町二丁目11番1号 株式会社エヌ・ティ・ティ・ドコモ内
 (72)発明者 ダニエル エイ バーネット
 東京都港区六本木1丁目4番30号 第25森ビル 日本ルーセント・テクノロジー株式会社内

(72)発明者 フラン バンケン
 オランダ、フィツエン、ポストブス 18、
 ポッターシュトラート 45、HE-3-04
 (72)発明者 アンドリュー ジョナサン ベンネット
 グレート・ブリテン及び北部アイルランド
 連合王国、スウェーデン、ビジネスパーク、オプティマス ウィンドミル ヒル
 1/125
 Fターム(参考) 5B076 AB10 AB17 BB06

* NOTICES *

Japan Patent Office is not responsible for any damages caused by the use of this translation.

1. This document has been translated by computer. So the translation may not reflect the original precisely.
2. **** shows the word which can not be translated.
3. In the drawings, any words are not translated.

DETAILED DESCRIPTION

[Detailed Description of the Invention]

[0001]

[Field of the Invention] It is the opening API (it is described as OpenAPI Open Application Programming Interface and the following.) aiming at this invention being exhibiting outside network functions which the Telecom carrier network's has, such as a call control function and an e-mail function, and things other than the Telecom carrier accessing, and enabling development of additional service application. It is related with a system.

[0002] Especially this invention relates to the service control equipment and the approach of arbitrating so that two or more service applications can operate simultaneously about the application using OpenAPI.

[0003] Furthermore, this invention relates to the service control equipment and the approach of offering the structure which conceals the distributed configuration of the OpenAPI server existing [two or more] to the application using OpenAPI.

[0004]

[Description of the Prior Art] When the user of the Telecom service uses service, the case where the service which service provision conditions are the same, or resembled closely is used can be assumed. It is for example, like answering machine service and a call forwarding service. In that case, the Telecom carrier which is the provider of the Telecom service needs to analyze the function and offer conditions of two or more services which an applicable user uses, and needs to offer the most suitable service preferentially. Conventionally, the approach of making fixed at the exchange on a network, when the judgment has grasped the function and offer conditions of all services in advance was taken.

[0005] By OpenAPI, the structure which can be notified in advance dynamically is provided with the conditions which the application using service makes start self-application to the Telecom carrier network to such an approach of making fixed. When the starting conditions notified in advance are filled with OpenAPI using this structure, the Telecom network starts that service. Therefore, while the fixed judgment processing for offering the most suitable service on the Telecom network preferentially becomes unnecessary, the sequential analysis of the service starting conditions dynamically notified from two or more service applications is carried out, and the judgment processing for offering the most suitable service preferentially is needed.

[0006] Moreover, by the OpenAPI specification, OpenAPI application and an OpenAPI server need to recognize each other to 1 to 1. Therefore, when two or more OpenAPI servers exist in the Telecom carrier network, OpenAPI application must recognize two or more OpenAPI servers of all separately. Moreover, in case OpenAPI application uses API for an OpenAPI server and notifies a demand, it must be determined appropriately to which server of two or more OpenAPI servers the demand is notified.

[0007]

[Problem(s) to be Solved by the Invention] As mentioned above, in OpenAPI, there is a limit that only one can register into the Telecom carrier network the service application which has the same service starting conditions. For example, if it says about answering machine service and a call forwarding

service, only any one can register service starting conditions into the Telecom carrier network. Under such a limit, many services cannot be employed flexibly.

[0008] Moreover, when two or more OpenAPI servers exist in the Telecom carrier network, in order to determine an OpenAPI server with suitable OpenAPI application, it is necessary to get to know the distributed configuration of an OpenAPI server. That causes complication of the service logic which OpenAPI application has, and checks the increase in efficiency of service development.

[0009] Then, the object of this invention is to offer the Sir building control unit and approach in the OpenAPI system simultaneously received by the Telecom carrier network, even if two or more service applications have what kind of service starting conditions.

[0010] Other objects of this invention are preferentially to offer a Sir building control unit and an approach. [in / for the suitable service for two or more service applications / the OpenAPI system which can be offered]

[0011] The object of further others of this invention has the application using OpenAPI in offering the Sir building control unit and approach in the OpenAPI system which makes service provision possible, without getting to know the distributed configuration of an OpenAPI server, when operating on the Telecom carrier network which consists of two or more OpenAPI servers.

[0012]

[Means for Solving the Problem] In order to attain the above-mentioned object, an opening API client and a network element usable [both of a server] and accessible are required in an opening API network. Although this network element needs for both opening API client and server, and a communication link to be possible, it should not know that these are communicating with the network element concerned. All the additional functions prepared in order to strengthen an OpenAPI specification, or in order to hide some of that imperfection are implemented in this network element.

[0013] The service control equipment by the 1st viewpoint of this invention Application using Opening API (Open Application Programming Interface) (it is hereafter called an opening API client.) In the opening API system containing the opening API server which operates as a part of Telecom network It is prepared in said Telecom network as a deputy of said opening API server to said opening API client. The memory which stores service starting condition registration / modification / deletion demand which received from said opening API client, An analysis means to analyze a service starting condition registration condition from at least one service starting condition registration / modification / deletion demand already stored in said memory, It carries out having a means to distinguish the service starting condition portion which modification does not produce, the service starting condition portion which should be changed, and the service starting condition portion which should be deleted, based on service starting condition registration / modification / deletion demand which received, and said service starting condition registration condition as the description.

[0014] When it should be judged and notified whether it should notify to said opening API server based on said distinction as a deputy of said opening API client to said opening API server, a service starting condition registration / modification / deletion demand constitutes from the service starting condition portion which said modification of said does not produce, said service starting condition portion which should change, and said service starting condition portion which should delete, and it carries out having further an advice judging means notify to said opening API server as the description.

[0015] The service control equipment by this invention is formed in said Telecom network as a deputy of said opening API client to said opening API server. If advice of service starting condition formation is received from said opening API server by service starting conditions being satisfied on said Telecom network A specific means to specify at least one opening API client which performed service starting condition registration corresponding to the advice of service starting condition formation concerned, When two or more opening API clients are specified by said specific means, it is characterized by having a means to determine the priority which notifies the purport that service starting conditions were satisfied to all the opening API client of them.

[0016] As a deputy of said opening API server to said opening API client, when only one opening API client is specified by said specific means, advice of the purport that said service starting conditions were

satisfied is sent to the opening API client concerned, and when two or more opening API clients are specified by said specific means, it has further the advice means which carries out sequential sending according to said priority in advice of the purport that said service starting conditions were satisfied. [0017] Said advice means is characterized by what advice of the purport that said service starting conditions were satisfied is sent for to the opening API client of the following priority, when the opening API client concerned to a response cannot be found to advice of the purport that said service starting conditions sent to one opening API client were satisfied.

[0018] Said opening API client and an opening API server are characterized by operating as deputy equipment with a standard OpenAPI specification, without changing an interface in any way.

[0019] As mentioned above, even if much unrealizable service application has what kind of service starting conditions, by the standard OpenAPI specification, registration of all service applications is simultaneously attained in the Telecom carrier network by adding the service control equipment by this invention to the system which consists of the servers and client applications based on a standard OpenAPI specification. Moreover, it becomes possible to perform the judgment for offering the most suitable service preferentially.

[0020] The service control equipment by the 2nd viewpoint of this invention Application using Opening API (Open Application Programming Interface) (it is hereafter called an opening API client.) In the opening API system containing two or more opening API servers which operate as a part of Telecom network As a deputy of two or more of said opening API servers to said opening API client A receiving means to receive either service starting condition registration / modification / deletion demand and a call generation demand from said opening API client, It is characterized by having an API distribution means to notify the demand which received with said receiving means to either of said two or more opening API servers as a deputy of said opening API client to said two or more opening API servers.

[0021] said two or more opening API servers -- said Telecom network top -- one -- although it is plurality physically -- logical -- one and 2 -- 3 which is plurality physically and is also logically divided into plurality -- it is distributed by said combination of 1 and 2, and the pattern of *****, and said API distribution means is characterized by determining one opening API server which should distribute said demand.

[0022] As mentioned above, it becomes possible to separate the static relation between an OpenAPI client and an OpenAPI server by adding the service control system by this invention to the system which consists of the servers and client applications based on a standard OpenAPI specification. This means that the demand of the degree from the same OpenAPI client may be transmitted to another OpenAPI server, even if it divides and the demand from a certain OpenAPI client is transmitted to one OpenAPI server. A service control system conceals a network topology to all OpenAPI clients.

[0023]

[Embodiment of the Invention] Hereafter, the operation gestalt of this invention is explained to a detail, referring to a drawing.

[0024] The 1st operation gestalt 1.1 service-control system chart 1 is a rough block diagram of the OpenAPI system using the service control system (SCS) by 1 operation gestalt of this invention. Here, the service control system 10 and the OpenAPI server 11 are formed in a network side, and the service control system 10 intervenes between two or more OpenAPI clients (application using OpenAPI) C1-CN, and the OpenAPI server 11. The interface of the service control system 10 is an interface based on an OpenAPI specification. Neither the OpenAPI clients C1-CN nor the OpenAPI server 11 needs to know existence of the service control system 10.

[0025] As mentioned above, in OpenAPI (for example, Parlay), only the interface at the time of a client and a server operating by 1 to 1 response is specified, and one client can require the control to one call. In such a system, since two or more clients may not know existence of a partner, the client from which plurality differs to control of one call may perform the demand which competes mutually. In an OpenAPI system, although such a demand that competes is forbidden, it becomes possible [requiring] with the Sir building control system 10 by this invention, without receiving an error for control of the same call using the OpenAPI specification with two or more same clients.

[0026] A service control program consists of a series of interlocking software modules as middleware, and can be written in code independent of a platform. So that it may mention later for example, a service control program The module which receives and stores service starting condition registration / modification / deletion demand, From a server the module which analyzes the already stored demand, the module which creates the demand which compares the newly received demand with the analysis result in this time, and is notified to a server, and advice of demand formation at the time of a carrier beam The module which determines in what kind of sequence advice of demand formation is transmitted to which client is included.

[0027] In addition, a service control process communicates between the OpenAPI clients C1-CN and the OpenAPI server 11 through the transport systems (for example, CORBA, WSDL, etc.) which are carrying out the current activity to the communication link with mutual usual transport system, i.e., OpenAPI client, and usual server. All the interfaces used on service control software are based on an OpenAPI specification.

[0028] 1.2) Acceptance and processing drawing 2 of the demand from two or more clients are a sequence diagram for explaining actuation of the service control system by 1 operation gestalt of this invention. If the service control system 10 accepts and processes service starting condition registration / modification / deletion demand from an OpenAPI client, they will be appropriately transmitted to the OpenAPI server 11. If a demand is first received from a client, the processed demand will be transmitted to the OpenAPI server 11.

[0029] 1.3) When the demand from two or more clients is received to control of the call of one advice of demand formation to two or more clients, and the call with which the service control system 10 fills the demand of a client is obtained from a network, which client needs to determine whether the advice should be received.

[0030] In drawing 2 , when service starting conditions are satisfied on a network, the service control system 10 receives a network event (here advice 11.0 of demand formation). If the advice 11.0 of demand formation is received, as for the service control system 10, which client will determine whether the advice 11.0 of demand formation concerned should be received first (step S12).

[0031] According to the OpenAPI specification, the client which received the event carries out a certain response. Therefore, if the advice 11.0 of demand formation is received, the OpenAPI client should return the response. when there is no response in any way (for example, the logic failure of a client, a network failure, etc. -- a cause), the service control system 10 sends the same event (advice 11.0 of demand formation) to the client of the following priority.

[0032] When all the client cannot be received to a certain event, either, the service control system 10 notifies an error to the OpenAPI server 11.

[0033] The 2nd operation gestalt 2.1 service-control system chart 3 is a rough block diagram of the OpenAPI system using the service control system (SCS) by other operation gestalten of this invention. Here, the service control system 20 and two or more OpenAPI servers S1-SM are formed in the network side. The service control system 20 intervenes between the OpenAPI client (application using OpenAPI) 21, and two or more OpenAPI servers S1-SM.

[0034] The service control system 20 is ***** as a proxy between the OpenAPI client 21 and an OpenAPI server (deputy equipment). The service control system 20 can receive registration / modification / deletion demand of event conditions or a call generation demand from the OpenAPI client 21. An OpenAPI server suitable as a distribution place is determined, and these demands are transmitted so that it may mention later.

[0035] From an OpenAPI client, the service control system 20 appears as an OpenAPI server. From an OpenAPI server, after 1OpenAPI client is attested, or the service control system 20 appears as a framework entity which requires a service interface instead of a framework, it appears as an OpenAPI client which requires call generation instead of 2, for example, other OpenAPI clients.

[0036] The interface of the service control system 20 is an interface based on an OpenAPI specification. Neither the OpenAPI client 21 nor the OpenAPI servers S1-SM need to know existence of the service control system 20.

[0037] Drawing 4 is rough hardware configuration drawing of the service control system 20. Besides the interfaces 201 and 204 for network connection, it has at least the processor 202 for executing the API distribution control program by this invention, and the memory 203 which stores required criteria data, such as service starting conditions.

[0038] An API distribution control program consists of a series of interlocking software modules as middleware, and can be written in code independent of a platform. For example, the API distribution control program contains the module which receives and stores service starting condition registration / modification / deletion demand or a call generation demand, the module which determines to which OpenAPI server it distributes by referring to service starting conditions so that it may mention later.

[0039] In addition, an API distribution control process communicates between the OpenAPI client 21 and the OpenAPI servers S1-SM through the transport systems (for example, CORBA, WSDL, etc.) which are carrying out the current activity to the communication link with mutual usual transport system, i.e., OpenAPI client, and usual server. All the interfaces used on the API distribution control software are based on an OpenAPI specification.

[0040] 2.2) Registration / modification / deletion demand for controlling the demand of the call and/or call generation which were received from the API distribution control OpenAPI client 21 may only be sent to a specific OpenAPI server by the former. For this reason, static mapping between an OpenAPI client and an OpenAPI server cannot be changed during activation. When based on the demand of the OpenAPI client of wanting you to tell whether some event conditions occurred, this static relation becomes for example, less desirable. Such inconvenience may be wiped away by introducing the service control system by this operation gestalt.

[0041] Drawing 5 is the sequence diagram showing an example of the API distribution control action of the service control system by this invention. The service control system 20 can receive registration / modification / deletion demand RQ 21.1 of event conditions or the call generation demand RQ 21.2 from the OpenAPI client 21.

[0042] Since the service control system 20 seems to have mentioned above as an OpenAPI server from the OpenAPI client 21, the network configuration of an OpenAPI server is hidden in the client. The service control system 20 is registered into the network of an OpenAPI server as a client.

[0043]

[Effect of the Invention] As explained above, even if much unrealizable service application has what kind of service starting conditions, by the standard OpenAPI specification, registration of all service applications is simultaneously attained in the Telecom carrier network by adding the service control system by the 1st operation gestalt of this invention to the system which consists of the servers and client applications based on a standard OpenAPI specification. Moreover, it becomes possible to perform the judgment for offering the most suitable service preferentially.

[0044] Furthermore, it becomes possible to separate the static relation between an OpenAPI client and an OpenAPI server by adding the service control system by the 2nd operation gestalt of this invention to the system which consists of the servers and client applications based on a standard OpenAPI specification. This means that the demand of the degree from the same OpenAPI client may be transmitted to another OpenAPI server, even if it divides and the demand from a certain OpenAPI client is transmitted to one OpenAPI server. A service control system conceals a network topology to all OpenAPI clients.

[Translation done.]

*** NOTICES ***

**Japan Patent Office is not responsible for any
damages caused by the use of this translation.**

1. This document has been translated by computer. So the translation may not reflect the original precisely.
2. **** shows the word which can not be translated.
3. In the drawings, any words are not translated.

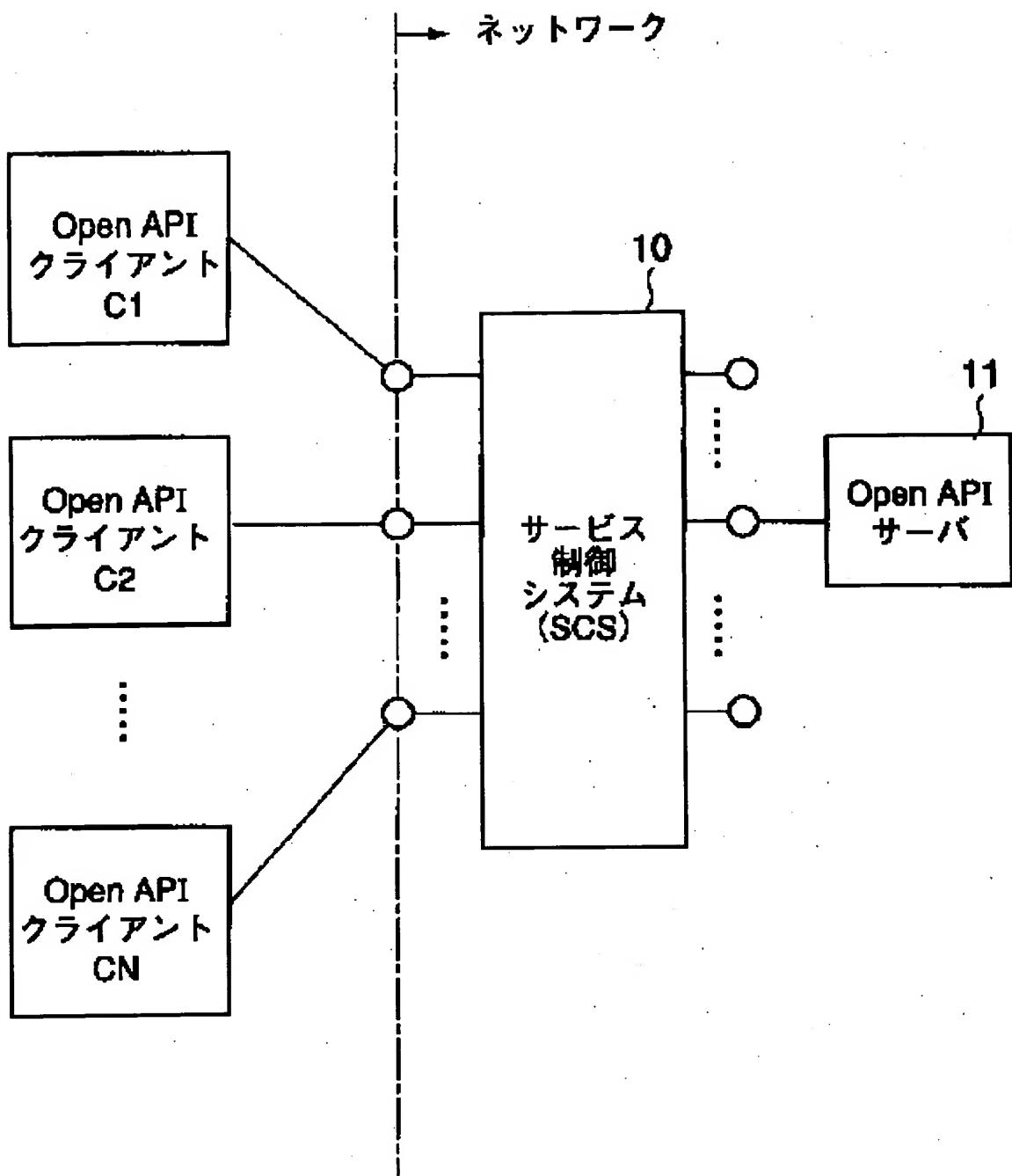
CLAIMS

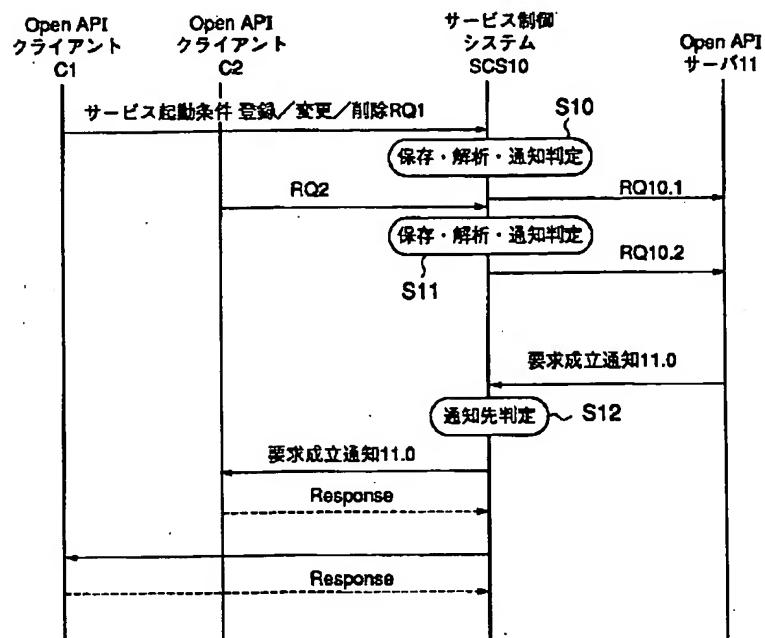
[Claim(s)]

[Claim 1] The application using Opening API (Open Application Programming Interface) (henceforth an opening API client), In the opening API system containing the opening API server which operates as a part of Telecom network It is prepared in said Telecom network as a deputy of said opening API server to said opening API client. The memory which stores service starting condition registration / modification / deletion demand which received from at least one opening API client, An analysis means to analyze a service starting condition registration condition from at least one service starting condition registration / modification / deletion demand already stored in said memory, The service starting condition portion which modification does not produce based on service starting condition registration / modification / deletion demand which newly received, and said service starting condition registration condition, Service control equipment characterized by having a means to distinguish the service starting condition portion which should be changed, and the service starting condition portion which should be deleted.

[Claim 2] As a deputy of said opening API client to said opening API server When it should be judged and notified whether it should notify to said opening API server based on said distinction The service starting condition portion which said modification does not produce, and said service starting condition portion which should be changed, Service control equipment according to claim 1 characterized by having further an advice judging means to constitute service starting condition registration / modification / deletion demand from said service starting condition portion which should be deleted, and to notify to said opening API server.

[Claim 3] The application using Opening API (Open Application Programming Interface) (henceforth an opening API client), In the opening API system containing the opening API server which operates as a part of Telecom network It is prepared in said Telecom network as a deputy of said opening API client to said opening API server.



Drawing selection drawing 2 

[Translation done.]

